Shadow Mapping

Anees Ur Rehman, Jesh Amera, Thuc Nguyen
aneesur.rehman001@umb.edu,yeshiwas.amera001@umb.edu,dangluong.nguyen001@umb.edu
University of Massachusetts Boston

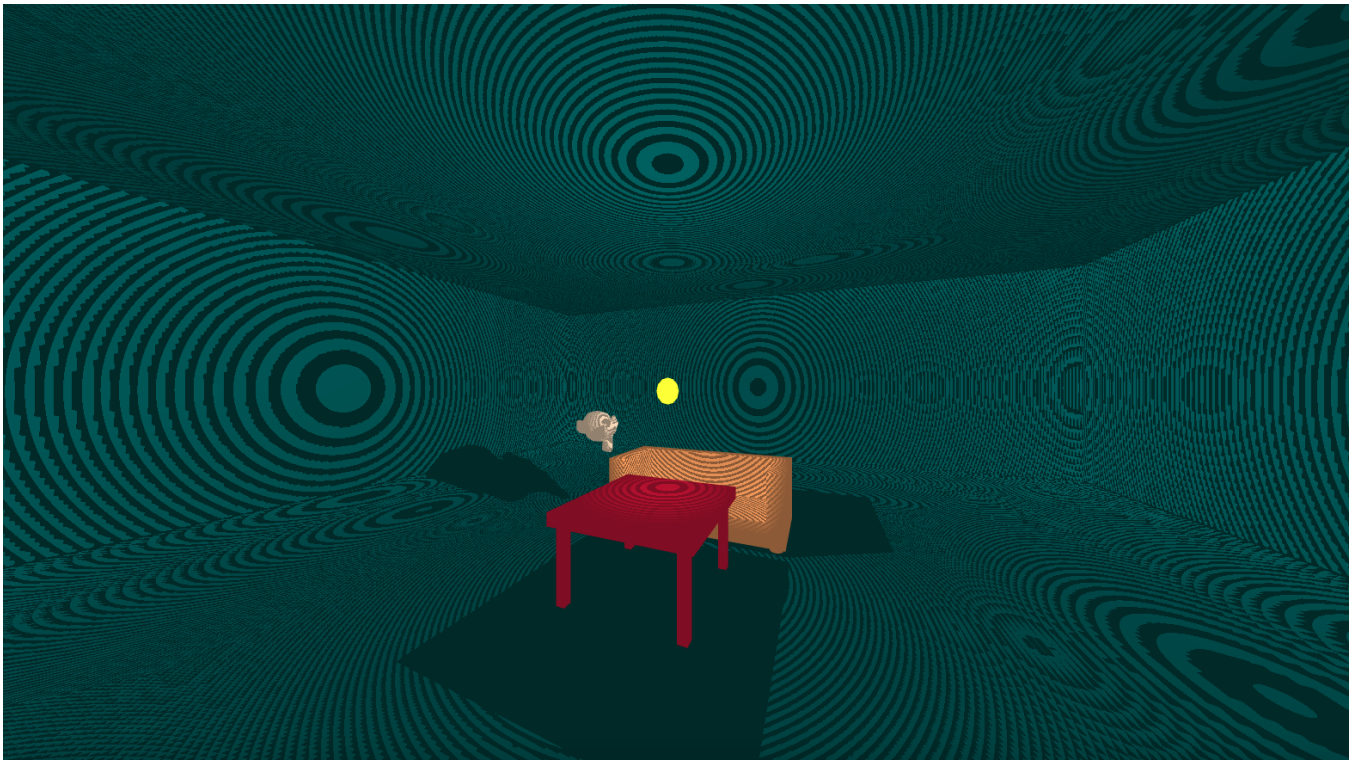

Figure 1: Shadow with the ripple effect

ABSTRACT

Shadows are a very important part of the real world and without them things would look a bit off. In graphics applications, shadows provide a much more lifelike depth of knowledge about a particular scene of objects or 3D models. Our 3D shadow mapping project begins with an animated 3D Visualization of Shadow mapping of certain objects placed in the room . One can navigate using the "W," "A," "S," and "D" also up, down, right and left arrows can be used. After the 3D objects rendered, it is easy to visualize and analyze the nature of the shadow in a three-dimensional space. Our project is based on the frameworks of WebGL

CS460, Fall 2020, Boston, MA

ACM ISBN 1337.
https://CS460.org

KEYWORDS

WebGL, Visualization, 3D, Shadow, mesh, OpenGL, Algorithms, Texturing, Shader, Ripple effect.

ACM Reference Format:
Anees Ur Rehman, Jesh Amera, Thuc Nguyen. 2020. Shadow Mapping. In *CS460: Computer Graphics at UMass Boston, Fall 2020.* Boston, MA, USA, 3 pages. https://CS460.org

1 INTRODUCTION

In this project, our primary focus is examining the steps involved in achieving shadow mapping, optimizations, and extensions that have enhanced the pre-existing project. This shadow mapping project explains the nature of shadows in computer graphics. Particularly we have explored how shadows are created from 3D objects. The project focuses on the subject of real-time shadowing rendering through an image space technique called shadow mapping using z-buffer and shaders. We have achieved real-time shadow mapping and several optimization and additions made, including large mesh subdivision, light source, and texture masked shadows.

2 RELATED WORK

This project is a small extension of the Kamaron Peterson's shadow mapping.

3 METHOD

This project covers dynamic shadow mapping. In this project we are recomputing the shadows for every single frame. This is done by looking from the point of view of the camera, whatever is visible is lit by the fawn lighting model, and those objects that aren't visible are lit by the ambient light. This project is purely Webgl based. This project involves 3D objects as well as moving objects. In order to do the computations and also do them in a easier way, we used a preexisting library called gl-martix which has so many function to do all kinds of 3D related calculations.

3.1 Implementation

The actual scene was created using a tool called blender which eventually converts the scene into the JSON object which we used in our project.

The following code snippet was used to increase the size of the 3D model: -

```
mat4.scale(me.WallsMesh.world,
me.WallsMesh.world, vec3.fromValues(2,2,2));
```

The code below fixes up the shadow distortion: -

```
me.gl.getExtension("OES_texture_float");
me.floatLinearExtension =
        me.gl.getExtension("OES_texture_float_linear");
    if (me.floatExtension && me.floatLinearExtension) {
          for (var i = 0; i < 6; i++) {
              me.gl.texImage2D(
                me.gl.TEXTURE_CUBE_MAP_POSITIVE_X + i,
                   0, me.gl.RGBA,
                   me.textureSize, me.textureSize,
                   0, me.gl.RGBA,
                   me.gl.FLOAT, null
              );
          }
      }
```

3.2 Milestones

How did you structure the development?

3.2.1 Milestone 1. In the meeting we discussed presenting the shadow mapping of objects in the room with some mesh objects such as a rotating floating monkey, a floating light that moves back and forth, and the sofa.

3.2.2 Milestone 2. After the meeting we started looking for some examples to understand the concepts and how to implement those features.

3.2.3 Milestone 3. Initially we wanted to increase the size of the room which was very small, and it was only possible after understanding the code, which we were able to figure out by going through the important sections of the files. We also changed the texture color of the objects.Added features for fixing the distortion in the shadow of the objects which was present in the actual demo. We were also able to add the RIPPLE LIGHT effect in our code. The Ripple light doesn't affect the actual shadow of the objects, but creates the waves instead only on the surronding.

3.3 Challenges

Describe the challenges you faced.

- Challenge 1: The biggest challenge that we faced was to clear up the distortion in the shadows of the object that the light was making.
- Challenge 2: The room was very small initially. In-order to make the room bigger we had to understand the gl-matrix library because the author based the original code on that library. But, later on we were able to figure out which function we are supposed to be used and what the parameters.

4 RESULTS

The light is casting almost perfect shadows of the object as shown in the image 3. The image 2 shows the initial shadow problem in the project.

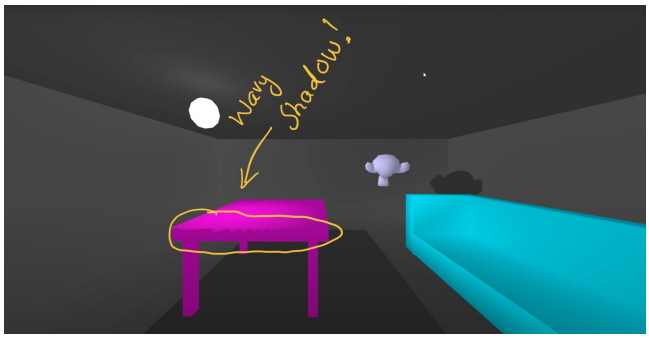

Figure 2: An example image of shadow distortion.

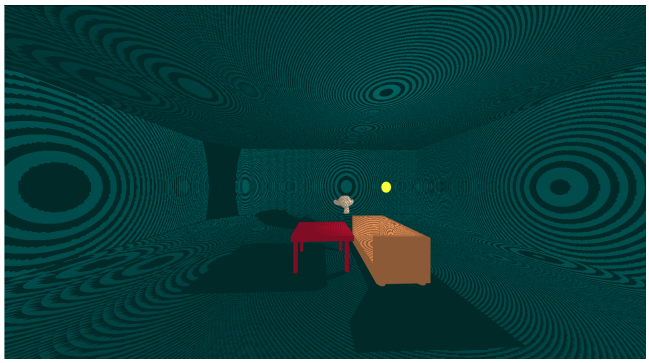

Figure 3: An example image.

Or you could add tables (see Table 1 - maybe with some timings?).

Table 1: Some example table

Device	Performance
iPhone	60 FPS
Android	60 FPS
Old Macbook	10 FPS

5 CONCLUSIONS

We enjoyed a lot by doing this project. We nicely worked in a team together and learned a lot by exchanging ideas and concepts. Working in a team enables you to communicate with your team members and get the job done. We hope that our project will make it to the favorite projects list of this semester.

REFERENCES

HEIDRICH, W. 1999. High-quality Shading and Lighting for Hardware-accelerated Rendering. PhD thesis.

KILGARD, M. J. 1999. Improving shadows and reflections via the stencil buffer. Published online at developer.nvidia.com.

KILGARD, M. J. 2001. Shadow mapping with today's opengl hardware. published online at developer.nvidia.com.

CROW, F. C. 1977. Shadow algorithms for computer graphics. In Computer Graphics (Proceedings of SIGGRAPH 77), vol. 11, 242–248.

Source of the code which we extended: -

https://github.com/sessamekesh/IndigoCS-webgl-tutorials/tree/master/Shadow%20Mapping